Fast Multipole Method for Nonlinear, Unsteady Aerodynamic Simulations

Abu Kebbie-Anthony¹, Nail Gumerov², Sergio Preidikman³, Balakumar Balachandran⁴, and Shapour Azarm⁵
University of Maryland, College Park, MD, 20742

The authors study the use of the Fast Multipole Method (FMM) for accelerating an aeroelastic simulator, comprised of the Unsteady Vortex Lattice Method (UVLM) for fluid dynamics simulations, and the Finite Element (FE) method for structural dynamics simulations. The FMM is integrated with UVLM based simulator. This accelerated UVLM model will in turn be used to accelerate the aeroelastic simulator. Considering the joined-wing SensorCraft, progress made thus far is reported in this work. The FMM algorithm is applied to a UVLM aerodynamic model for a large aspect ratio, planar, rectangular lifting surface and the results obtained on the computational cost reduction are presented. The current approach has broad applicability for the study of aerodynamic and aeroelastic responses of aircraft systems.

I. Introduction

The mission effectiveness of Unmanned Aircraft Systems (UAS) can be influenced by unforeseen system responses and environmental conditions. Through these unexpected events, the UAS could suffer structural damage that will make the system more susceptible to aeroelastic instabilities. An UAS subjected to aeroelastic instabilities long enough will tend to fail depending on the material properties and design of the aircraft. Current designs of surveillance and observation platforms and data collection drones with high aspect-ratio and highly flexible wings, for example, the joined-wing SensorCraft [1,2], bring forth the necessity for considering flexible aeroelastic loads in order to provide realistic estimates of the aerodynamic performance and the aeroelastic response of the aircraft. Linear aeroelastic models, although amenable to fast computations, are not adequate to capture post-critical aeroelastic behavior and these models cannot be used to estimate reliable margins for aeroelastic instabilities; that is, to generate a safe maneuvering envelope. In this regard, there is an urgent need to develop a set of fast, robust, accurate, and reliable prediction methods based on fully coupled aerodynamics, structural dynamics, control systems, and nonlinear analysis.

Typically, an aeroelastic model [3,4] is developed through the integration of a structural dynamics model, usually a Finite Element (FE) model, an aerodynamic model, and a procedure for the inter-model connection; that is, for transferring information between the aerodynamic and the structural dynamics simulators. The bi-directional exchange of information between the simulators for aerodynamics and structural dynamics is referred to as a co-simulation strategy [4-6], which will be elaborated on more in a later section.

For the aerodynamics, the Unsteady Vortex Lattice Method (UVLM) [3,4,7] can be used to compute the flowfield around a body and forces acting on this body. This method can be used to model flows over 3D lifting and non-lifting surfaces undergoing arbitrary time-dependent deformations and any motion in space. UVLM accurately describes aerodynamic interference among bodies. The bound-vortex sheets, which model the boundary layers, are replaced by a lattice of short, straight vortex segments with circulation $\Gamma(t)$. These segments are used to divide the body surface into a number of elements of area (the so-called panels that generally are nonplanar). The model is completed by joining free vortex lines, representing the wakes, to the bound-vortex lattice along the separation edges, such as the wing-trailing edges and the wing-tips. The locations at which separations occur are input data. However, in the wakes, the positions of the vortex segments and their circulations are determined as a part of the solution. As time is evolved, the number of vortex segments associated with the wake grows in proportion to the number of vortex segments on the trailing edge and wing-tips of the considered aircraft’s wing.
In order to calculate the velocity field using the UVLM, which involves the Biot–Savart law [7], the influence of \( N \) discrete finite vortex segments with constant circulation must be computed; the associated computational cost can be ordered as \( O(N^2) \). It should be noted that due to the convection of the wake, the value of \( N \) goes up as time is evolved. In an effort to distribute this computational cost, Chabalko et al. [8] and Chabalko and Balachandran [9,10] implemented two-dimensional vortex interactions on a Graphics Processing Unit (GPU) and examined how GPU computing can be used to speed up computations of vortex interactions. While effective in reducing the computational work load, speedup gained from GPU computing is hardware dependent.

In this work, an algorithmic approach for attaining a reduction in the vortex interactions computations is explored. The Fast Multipole Method (FMM) has been identified as one of the ten algorithms with the greatest influence on the development and practice of science and engineering in the 20th century [11]. With the FMM, the computational cost of the \( N \)-body problem is reduced from \( O(N^2) \) to \( O(N) \) or \( O(N \log N) \), which is essential when \( N \) becomes large. The FMM algorithm was first developed in 1987 by Greengard and Roklin [12] to calculate gravitational and electrostatic potentials. The FMM was further enhanced by Carrier et al. [13] for the evaluation of potential and force fields in systems involving large numbers of particles. In the work reported in references [14, 15], the FMM is applied towards the evaluation of Laplace’s equation governing fluid flow. The FMM has been shown to be helpful with the approximation of fluid flow governed by the Navier-Stokes equations through vortex methods. Gumerov and Duraiswami [16] implemented the FMM for simulations based on vortex methods via the Lamb-Helmholtz decomposition. Recently, the application of the FMM algorithm with Direct Vortex Methods (DVM) for free domain and periodic problems was presented by Ricciardi et al. [17]. The acceleration gained through the application of the FMM can be further enhanced with the use of GPUs [18-20].

In the current work, as mentioned earlier, the motions of aircraft systems with long, slender wings are of interest. As reported in the group’s prior work, systems with flexible wings can exhibit complex motions [21]. With the long, flexible wings of the SensorCraft, pilots need to take into account nonlinear aeroelastic instabilities when tailoring flight configuration options. A decision support system will help pilots tailor such flight configuration options. The Dynamic Data-Driven Application Systems (DDDAS) paradigm is being utilized with the accelerated aeroelastic simulation and meta-models to develop a decision support system. The DDDAS paradigm can be used to realize a framework in which measurement data, such as sensors or GPS data, are collected for a given physical system and are used to dynamically update a computational model of the physical system. Darena [22] introduced the DDDAS concept in 2004. The decision support system will enhance the autonomy of an UAS under the effects of nonlinear phenomena. Work by Farhat and Amsallem [23] and Allaire et al. [24] utilize the DDDAS paradigm to predict the failure and degradation in UAS and tailoring planned missions to best suit the remaining capabilities of the aircraft. With the DDDAS paradigm, there is an emphasis on fast simulations. With the reduction in computational expense, more simulations can be executed for a shorter amount of time, which leads to more available simulation data. Given the variety of scenarios that arises due to unforeseen events in the system and environment, more simulation data can be utilized to predict the expected response of the system. Furthermore, meta-models can be developed to approximate the nonlinear models, reducing the number of execution calls of the full aeroelastic simulator for decision support. Through this work, the authors will show how accelerated aeroelastic simulations can be realized by using the FMM algorithm.

Although in the literature, it has been reported that the use of the FMM can computationally accelerate aerodynamic simulations, an examination of the acceleration of the UVLM using the FMM has not been carried out thus far. This is addressed in the current work, wherein the authors implement the FMM for accelerating the UVLM based aerodynamic simulator. Integrating the accelerated UVLM model with a FE structural model, an aeroelastic simulator can be developed through a co-simulation strategy. The acceleration of the nonlinear and unsteady aerodynamic simulator will subsequently also allow for acceleration of the complete aeroelastic simulator. Application of the accelerated UVLM model to a large aspect ratio, planar, rectangular lifting surface is considered as a benchmark example.

II. Problem Statement

The need for accelerated aeroelastic simulations arises, when considering the uncertainty in the system environment. UAS system performance is limited by uncertainty in physical responses and changes in environmental conditions. Aircrafts designed with large aspect ratio, flexible, thin wings, are more susceptible to aeroelastic instabilities. This calls for nonlinear aeroelastic models to capture flutter and other nonlinear aeroelastic instabilities, such as aeroelastic buckling, which can greatly reduce the operational lifetime of the aircraft. However, these aeroelastic models are computationally expensive.
Aeroelasticity [25] refers to the study of the interaction amongst inertial forces, elastic forces, and aerodynamic forces. Through the construction of complicated geometries made of interconnected inertial and elastic elements, the FE structural model can be used to predict how a system reacts to real-world forces, vibration, heat, fluid flow, and other physical effects. In Computational Fluid Dynamics (CFD), forces that a fluid exerts on a rigid or an elastic body are simulated to obtain aerodynamic forces. Between these two modeling techniques, all three corners of Collar’s triangle are incorporated and aeroelastic phenomena are captured. These aeroelastic effects are created as a fluid flow exerts a force on a body, the body deforms under the loads, and the change in geometry in turn affects how the fluid flows. Due to this feedback loop, aeroelastic simulations tend to be difficult and time consuming.

To address this issue, here, the FMM is integrated with the UVLM aerodynamic model to accelerate the aeroelastic simulator. In the considered aeroelastic simulator, the major portion of the computational expense is associated with computation of the unsteady, nonlinear aerodynamic loads. The interaction between the aerodynamics and structural dynamics simulators in the aeroelastic system is illustrated in Figure 1. The solid line boxes are portions from the FE structural model of the aeroelastic simulator, while dotted line boxes are portions of the UVLM aerodynamic model. Given the initial conditions of the structure and the fluid flow, the wake is convected and frozen. This means that after the new positions of the fluid particles are computed through the evaluation of the flow velocity field, the computed fluid particles are stationary for the remainder of the time step. Next, the aerodynamic loads, such as the lift and pressure drag, are calculated. These loads are then transferred to the structural model to predict displacements and accelerations of the system. The predicted displacements and accelerations are used to update the aerodynamic grid and the aerodynamic loads. The transference of the aerodynamic loads and structural displacements and accelerations is repeated until the responses determined through the aerodynamic and structural models converge. In the UVLM aerodynamic model, convection of the wake and calculation of the aerodynamic loads require evaluation of the flow velocity field. The evaluation of this velocity field requires the computation of the influence of the \( N \) discrete vortex elements with constant circulation. This presents an \( N \)-body problem whose computational cost is of the order \( O(N^2) \).

To form the wake, vorticity is shed and convected, or moved, with the fluid flow at the local flow velocity. The wake influences the aerodynamic forces, and consequently, the aerodynamic loads are history-dependent. Thus, convection of the wake adds more discrete vortex segments for the evaluation of the velocity field. The presence of more discrete vortex segments in the wake can lead to more accurate determination of aerodynamic loads. However, as the wake continues to grow over time, the number of operations required for the velocity field computation become significantly high, which adds extra computational cost to the aeroelastic simulations. Thus, given an aerodynamic mesh with a large number of elements, attaining useful information from the aeroelastic simulations in a timely manner is nearly impossible. In Figure 2, the contributions of the aerodynamic model processes relative to the total computational time of the aeroelastic simulator are depicted. The computational time required to compute the

![Figure 1: Aeroelastic scheme: coupling between structural dynamics (solid line boxes) and aerodynamics (dotted line boxes) stages.](image-url)
structural response; that is, to numerically integrate the structure’s equations of motion, is neglected because it is dwarfed by the aerodynamic calculations. For similar reasons, the processes of the aerodynamic model involving the evaluation of the vortex circulations and the calculations of the aerodynamic loads are negligible. Notice that as the number of sources and receivers (field points and discrete vortex segments) increase, the wake convection process becomes the most computationally expensive portion of the aeroelastic simulator. This emphasizes the need for the FMM in the convection of the wake to accelerate the aeroelastic simulator. With the FMM algorithm, the number of pairwise field point computations needed to evaluate the velocity fields is reduced through approximations instead of direct calculations. The use of approximations in the FMM algorithm reduces the accuracy of the calculations but the level of accuracy can be easily controlled. Through this work, the authors will do the following: i) illustrate the use of the FMM algorithm to accelerate UVLM aerodynamic simulator; ii) discuss how the FMM accelerated UVLM integrated with a FE structural model will subsequently accelerate the aeroelastic simulator; and, iii) present the results of the FMM accelerated UVLM aerodynamic model for a large aspect ratio, planar, rectangular lifting surface.

III. Simulation Strategy

A. Co-Simulation Strategy

Aeroelastic computations are carried out by using a co-simulation strategy (e.g., [4]). For this work, in the aeroelastic simulator, the authors use the UVLM to predict the aerodynamic loads on the lifting surfaces. Additionally, this solver is coupled with a structural dynamics simulator to capture aeroelastic phenomena. Co-simulation here refers to subdividing a system with coupled physics into subsystems that are simultaneously simulated and numerically combined with a suitable exchange of states at predefined time instances to account for the strong coupling.

In Figure 3, the steps involved in the co-simulation process for a SensorCraft wing in an airflow are depicted. At the initial step, the coupled system (structure in airflow) is represented by the system, wherein the state is given by \(\begin{pmatrix} x(t) \\ y(t) \end{pmatrix}\). Here the state vectors, \(x(t)\) and \(y(t)\), are the displacements and accelerations associated with the SensorCraft wings and the aerodynamic loads associated with the fluid flow, respectively. In the next time step, the system is partitioned into two subsystems as follows:

\[
\frac{d}{dt} \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \end{pmatrix} = \begin{pmatrix} f_1(x(t),y(t)) \\ f_2(x(t),y(t)) \end{pmatrix} \Rightarrow \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \end{pmatrix} = \begin{pmatrix} f_1(x(t),y(t)) \\ f_2(x(t),y(t)) \end{pmatrix}.
\] (1)
The final step of the co-simulation process involves the exchanging of information between the two subsystems bi-directionally. Since the subsystems are simulated separately, predictions are needed for the unknown state vectors in the opposite subsystems. To simulate the displacements and accelerations, \( \dot{x}(t) \), of the aircraft, a prediction \( v(t) \) is needed for its \( y(t) \) input. Similarly, to simulate the aerodynamic loads, \( y(t) \), associated with the fluid flow, a prediction \( u(t) \) is needed for its \( x(t) \) input. Substituting the predictions, \( u(t) \) and \( v(t) \), into system (1), the system can be written as

\[
\begin{align*}
\dot{x}(t) & = f_x(x(t), v(t)) \\
y(t) & = f_y(u(t), y(t))
\end{align*}
\]

(2)

In this sense, the displacements and accelerations of the aircraft’s wings are calculated by using the predicted aerodynamic load, \( v(t) \), and aerodynamic loads are computed by using the predicted displacements and accelerations, \( u(t) \). More information on co-simulations of complex systems can be found in references [4-6]. Co-simulations can consist of any number of subsystems but in this effort, two subsystems are involved in the co-simulation strategy. The two subsystems are computational implementations of models intended for the aerodynamics and the structural dynamics of the system.

![Figure 3: Three parts of co-simulation process for a SensorCraft wing in an airflow. a) Coupled system. b) Partition into two dynamic subsystems. c) Bi-directional exchange of information between subsystems.](image)

i. Structural Model: Finite Element Method

The first subsystem, Simulator 1, is the UAS structural model, obtained through the FE method. The beams model of the forward right wing and the rear right wing of the representative SensorCraft used for the structural model is shown in Figure 4(a). Nodal points are represented by circles distributed along the centerlines of both wings. In Figure 4(b), the flowchart of Simulator 1 is presented. The mass matrix, \( M \), and stiffness matrix, \( K \), are computed from the material and geometric properties of the aircraft wings and the load vector, \( F(t) \), is obtained from the UVLM aerodynamic model (Simulator 2). The computed modal displacements, \( q(t) \), and modal accelerations, \( \ddot{q}(t) \), are corrected by using the aerodynamic mesh until convergence between the structural and aerodynamic meshes is obtained.

ii. Aerodynamic Model: Unsteady Vortex Lattice Method

The second subsystem, Simulator 2, is the aerodynamic model, obtained with the UVLM. The aerodynamic mesh, or the vortex-lattice, used for the aerodynamic model is shown in Figure 5(a). The refinement level of the mesh correlates with the computational cost of the aerodynamic model. The flowchart of Simulator 2 is shown in Figure 5(b). In the model used, the wake is convected at the beginning of each time step and then held stationary for the remainder of the time step. The aerodynamic coefficient matrix and velocity fields are computed separately and then used to solve for the circulation of the discrete vortex segments. From the circulations, the aerodynamic loads are calculated and transferred to the structural model to solve for the modal displacements and accelerations.

The UVLM is a surface vorticity model that is used to accurately approximate the physical reality of a very large Reynolds number, fully attached, flow. The infinitesimally thin layers of vorticity may be viewed as the infinite Reynolds number approximation to the actual boundary layers. In Figure 6, the authors show a typical closed loop of
vortex segments, or vortex ring, where the circulation of the individual vortex segments is the same as the circulation of the vortex ring, and how these vortex segments contribute to the velocity at the field point $P$. The velocity at the field point $P$ and at time $t$, $V_i(P,t)$, associated with a discrete segment of a straight line vortex, $L_i$, $i=1,\ldots,N$, of circulation strength, $\Gamma_i(t)$, can be evaluated according to the Biot-Savart law:

$$Mq(t) + K\ddot{q}(t) = F(t)$$

Figure 4: a) Beam element representation of the right forward and rear wings of a representative joined-wing aircraft. b) Flowchart of Simulator 1.

Figure 5: a) Mesh used for aerodynamic model of joined-wing aircraft. b) Flowchart of Simulator 2.
\[
V_i(P, t) = \frac{\Gamma_i(t)}{4\pi} \frac{L_i \times r_j}{\|L_i \times r_j\|_2} \left[ L_i \cdot (\hat{e}_1 - \hat{e}_2) \right] = A_i(L_i, P) \Gamma_i(t)
\]

Here, \(r_1\) and \(r_2\) are the position vectors from the endpoints of the vortex segment to the field point \(P\) and \(\hat{e}_1\) and \(\hat{e}_2\) are unit vectors in the directions of \(r_1\) and \(r_2\), respectively. The velocity field at point \(P\) can be computed as the summation of velocity fields associated with the field point, \(P\), and the discrete vortex segments \(L_{i=1,N}\):

\[
V(P, t) = \sum_{i=1}^{N} V_i(P, t) = \sum_{i=1}^{N} A_i(L_i, P) \Gamma_i(t)
\]

The velocity field needs to be evaluated at \(M\) field points, \(P_j, j = 1, \ldots, M\) which leads to a computational cost of \(O(NM)\). The computational cost of the simulation rises significantly with the number of field points and vortex segments. For a system on the scale of a full aircraft, these computations become intractable. Such computational expense motivates the need for the FMM to accelerate the velocity field calculations. The FMM can be used to reduce the computational cost from \(O(NM)\) to \(O(N + M)\), thus, making the computational runtime more practical.

**B. The Fast Multipole Method**

The FMM is a hierarchical algorithm, which can be used to speed up matrix-vector products. The main idea of the FMM is to split system (4) into near-fields and far-fields interactions. This is done through the decomposition of the dense matrix into sparse and dense parts as

\[
V(P_j, t) = \sum_{i=1}^{N} A(L_i, P_j) \Gamma_i(t) = \sum_{i=1}^{N} A^{(spur)}(L_i, P_j) \Gamma_i(t) + \sum_{i=1}^{N} A^{(dov)}(L_i, P_j) \Gamma_i(t)
\]

where \(L_1, \ldots, L_N\) are the sources (vortex segments) and \(P_1, \ldots, P_M\) are the receivers (field points). The sparse matrix-vector product is performed directly, while the dense matrix-vector product is approximated via the use of data structures, the generation of multipole expansions, and the evaluation of local expansions. This means that the interactions between near-field pairs of field points and vortex segments are directly computed, while the interactions between the far-field pairs of field points and vortex segment pairs are approximated. The use of the hierarchical data structures (octree in our case) and translation operators enables computing of the dense matrix-vector product by using \(O(N)\) or \(O(N \log N)\) operations. The accuracy of the approximation is controlled by the length of the expansion series, or the truncation number, \(p\). For spherical harmonic basis, the number of terms in the expansions is \(p^2\), where the maximum degree of the spherical harmonics is \(p - 1\) (so the local expansions are harmonic polynomials of degree \(p - 1\) with respect to the Cartesian coordinates). For example, in the present study, \(p = 25\) was consistently
used, which shows that the far-field interactions were approximated by polynomials of the 24th degree in the neighborhood of the evaluation point, which in our case provided 7-9 correct digits in the computed values of the velocity. More details on the basics of the FMM can be found in references [12-15,18, 26]. In the present study, the FMM described and implemented in reference [16] is used, but without Lamb-Helmholtz decomposition (instead of two scalar potentials, three components of the vorticity were used). As it was found in reference [16], such an approach is slower than the newer method proposed in reference [16] by approximately 20% for the same truncation number. However, it was also found that the method with three components is more accurate for the same truncation number, so the actual gain in the speed for the same accuracy needs to be studied more carefully in future work.

It can be mentioned that a number of parameters, such as the truncation number, \( p \), clustering parameter, \( s \), or the number of levels in the octree controls the speed of the FMM. There are opportunities to “trade” accuracy for speed, by varying these parameters, which also can be optimized for the required accuracy. The FMM accuracy should be set based on the overall accuracy of the algorithm, and, in principle, should be determined by using data on the time marching schemes, the mesh size, accuracy of the fluid-structure interaction models, and other parts of the UVLM to have a stable and consistent overall approximation. In the present study, however, this has not been addressed. Furthermore, a complete optimization of the aerodynamic model was not performed, as such tuning can be a subject for separate investigation. The goal of the present study is to show the feasibility of the use of the FMM in the UVLM and estimate the accelerations of the working model that can be obtained by using the FMM for the most time consuming part of the algorithm. The results of this study are reported in the next section.

Finally, it should be noted that currently there exist different versions of the FMM for the Laplace equation in three dimensions in the form of open source codes and commercial software. Such packages can be used as “black-box” solvers, for integration with the core UVLM aerodynamic model without substantial modifications (one can run the FMM for scalar functions three times and compute the vorticity based on the computed gradients of the scalar components). While the performance for different FMM implementations can differ several times in any case, the FMM is scaled as \( O(N) \) or \( O(N \log N) \), which should bring substantial accelerations at large \( N \).

IV. Benchmark Application

The main purpose of this paper is to provide a basic comparative performance test of the FMM UVLM, with a basic implementation on a planar, rectangular lifting surface with a large aspect ratio. Wall clock times reported below were measured on an Intel® Xeon® CPU E3-1245 v5 (3.50 GHz) 8 core PC with 16 GB RAM. For the authors’ study, the planar, rectangular lifting surface has an aspect ratio of 16 and is subjected to a freestream velocity of magnitude
125 m/s, angle of attack $\theta^*$, and air density $1.255 \text{ kg/m}^3$. The FMM is used with double precision, the truncation number is set to $p = 25$, and the clustering parameter is set to $s = 200$. No specific value is set for the number of levels in the octree but rather this value is optimized given the number of sources and receivers.

In Figure 7, the results of the speed performance of the UVLM with and without the FMM is depicted. In the figure, the computational time for the evaluation of the velocity fields in the wake convection process of the UVLM is plotted against the number of sources and receivers. The plus sign markers are a measure of the wall clock time (in seconds) of the non-FMM accelerated UVLM, while the circular markers are a measure of the time for the FMM accelerated UVLM. In logarithmic scale, the slope of the dotted line is 2, which indicates the computational expense of the non-FMM accelerated UVLM velocity field evaluation is of $O(N^2)$, where $N$ is the number of sources and receivers. The slope of the computational expense of the FMM accelerated UVLM is indicated by the solid line, which has a slope of 1. Thus, with the FMM, the computational expense of the velocity field evaluation in the wake convection process in the UVLM is reduced from $O(N^2)$ to $O(N)$. When $N = 200,000$, the FMM accelerated UVLM took about 10 seconds to compute the velocity fields compared to the 15 minutes the non-FMM accelerated UVLM needed! This improvement in speed is achieved with no noticeable loss in accuracy. The maximum absolute error of the computed aerodynamic load coefficients of the FMM accelerated UVLM is of magnitude $10^{-7}$.

V. Concluding Remarks

In this paper, the FMM is implemented for accelerating the UVLM aerodynamic model. This is the first study wherein the FMM has been implemented in a nonlinear, unsteady aerodynamic simulator. The aerodynamic model has been used to study a planar, rectangular lifting surface with a large aspect ratio. Through this benchmark application, the authors have shown that the FMM can be successfully used to reduce the computational expense of the velocity field computations from $O(N^2)$ to $O(N)$. The significant accelerations allow for simulation domain sizes that were previously computationally intractable. Thus, modeling more complex systems, such as the joined-wing SensorCraft, becomes more practical. Additionally, as shown here, the errors introduced by the FMM approximations of far-field interactions have almost no effect on the simulations. Here, only the acceleration of the wake convection, which is the most computationally expensive portion of the aerodynamic model, is accelerated through the FMM in this paper. In the future, acceleration of other aspects of the computation in determining the aerodynamic loads will be explored and the benefits of this acceleration on aeroelastic computations will be examined.

Acknowledgments

Partial support received from the U.S. Air Force Office of Scientific Research DDDAS Program through Grant No. FA95501510134 and the National Science Foundation through Grant No. CMMI-1250187 are gratefully acknowledged.

References


